EXTRACTION AND TRACKING OF ORIENTATION CODED FEATURES BEING ROBUST AGAINST ILLUMINATION CHANGES

Yukiyasu Domae, Hidenori Takauji, Henrik Stier, Shun'ichi Kaneko and Takayuki Tanaka

Graduate School of Information Science and Technology
Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814, JAPAN
domae@ssci.ssi.ist.hokudai.ac.jp
http://ssc-lab.com/
Faculty of Mechanical Engineering and Transport Systems
Technical University Berlin
Straße des 17. Juni 135, 10623 Berlin, GERMANY

Commission III/3

KEY WORDS: feature extraction, tracking, orientation code, richness

ABSTRACT:

We present a feature extraction and tracking method which is based on pure rich texture analysis and has robustness under irregular conditions, such as illumination change, deformation of objects and so on. The method is composed mainly of two algorithms: Entropy Filtering and Orientation Code Matching (OCM). Entropy filter points up areas of images being messy distribution of orientation codes. The orientation code is determined by detecting the orientation of maximum intensity change around neighboring eight pixels. The areas being extracted by the entropy filter doesn't depend on edges and gradations of images and has robustness against illumination change and motion of objects. Therefore, we can say the areas have "pure rich texture" and are suitable for tracking based on visual information. And then, OCM, a template matching method using the orientation code, is applied to track templates being centered on the features and updates the templates each frame. OCM is speeded up by using the information of texture richness and motion of objects. Because of their large robustness, we can track the templates robustly under irregular conditions.

1 INTRODUCTION

Machine visions are artificial systems in real world environments, which are based on image processing, classifier technologies and/or control engineering, and they have been widely utilized in the fields of industrial inspection, environmental monitoring, robotic production lines, control and measurement systems. Robust feature extraction and matching in machine vision are the fundamental technologies for pre-processing of tracking and consistent identification of objects or targets in the scene, such as outdoor scenes, non-laboratory circumstances, and visual tasks handling fluctuating objects like humans. We need to consider how to design these robust feature handling technologies which should consist of some steps in order. The first step is acquiring or discovering visual cues or features from the dynamic scene, which has been addressed, for instance, in computer vision for a long time. Among these discussions, edge detection has been one of the most commonly used solutions, therefore, many edge detectors (I.Sobel, 1978, J.Canny, 1986) have been proposed. In many cases, however, edges are easily affected by manually specified parameters in any implementations and also by change in physical conditions, such as illumination and shading. Then, in many researches, non-edge features have been designed for practical applications. Moravec operator (Y.Lamdan and H.J.Wolfson, 1977) extracts features by evaluating the variance of intensity in the neighboring pixels. Harris operator (C.Harris and M.Stephens, 1988) is based on the variation of local auto-correlation over different orientations. KLT algorithm (J.Shi and C.Tomasi, 1994) which is based on a model of affine image change is well-suited to extract features from video-sequences and tracks them well. They are well-known non-edge feature detectors, but they have some weakness in robustness against local illumination change such as shading or highlighting, which may cause troubles in identifying temporal consistency of spatially distributed individual features in the dynamic frames. Next, each feature has to be track consistently in video-sequences by using various algorithms. Visual tracking methods based on motion prediction, such as Kalman Filter (R.E.Kalman, 1960), which enables us to obtain rough trajectories of objects without high computation cost, have been suited for real-time applications like visual feedback and visual servoing. These methods, however, bear some difficulties in tracking accuracy and in the parameter estimation, such as estimated noise variance matrices. Kalman Filter basically can only deal with linear motions, therefore, it is difficult to track the objects doing the complicated motions rather than piecewise linear motions. Furthermore it is easily affected by change in characteristics of disturbance or noise, and then difficult to maintain the high tracking capability. Monte Carlo filtering, in particular, Condensation algorithm (M.Isard and A.Blake, 1996) is a well-known method to track rather complicated motions. It can track an object which have complicated motions and backgrounds, and be extended to multiple objects (D.Tweed and A.Calway, 2002), but needs the high computation cost to do many samplings for stable tracking. We should remember again all of these tracking methods depend on the precision and reliability of how to extract visual features/information in difficult scenes.

A problem of these methods is the existence of weakness in illumination change, occlusions and deformation of objects. We have developed Orientation Code Matching (OCM) (F.Ullah et al., 2001, F.Ullah et al., 2002), as an effective and efficient algorithm for robust template matching, in which a robustly coded image is made by reconstructing orientation information around neighboring eight pixels and then the similarity is evaluated by the sum of absolute differences between corresponding coded pixels. And then, the richness in OC coded images (H.Takauji et al., 2005) is developed as a feature detector which is robust for illumination change and can robustly select local areas of messy distributions in orientation coded images as robust and informative features for visual tracking. In this paper, we provide a comparative study with our OC-based feature extraction method and KLT algorithm, and propose a novel approach for autonomous definition and extraction of robust features and their renewal, which are based on OCM and the richness. And then, we apply them to problems of real-time tracking of objects in video-sequences. In the method of feature definition, richness distributions are filtered by an inverse Gaussian weighting in order to make the features distribute rather uniformly in the scene. In the process of feature renewal, we utilize the k-means clustering technique to sublime local texture cues to informative features for consistent and stable tracking. With real world dynamic scenes, we could obtain good performance of object tracking in spite of fluctuation or change in illumination and object posture.

The rest of the paper is organized in six sections. The details of feature extraction and tracking algorithm are described in Section 2 and 3. Then, in Section 4, we provide some experiments mainly for comparative evaluation of the OC-based feature extraction and KLT algorithm, results of tracking planar landmarks. In Section 5, an automatic feature definition and a renewal method are proposed, and the experimental performance of stable tracking by the proposed methods is shown. Finally, we conclude the paper with some remarks and comments on the future works.

2 FEATURES FOR TRACKING

2.1 OUTLINE

We give in this section a detailed fundamental of robust features used throughout this paper. Fig. 1 shows a flowchart of the feature extraction procedure. The intensities obtained by any vision sensors, which are represented in raw images in the figure, must depend on measurement conditions, such as illumination, reflection, and sensor conditions, therefore, they often cause difficulties in image processing tasks in practical situations. In order to limit their nuisance influence, we use Orientation code (OC) as a robust representation mainly for illumination fluctuation. Entropy filtering causing richness followed by a simple smoothing by a box filtering is introduced for evaluating how informative any local distributions of OC is. The combination of OC and the richness brings a stable feature analysis which is unaffected by inclusion of trivial cues, such as shadow, gradation, or unidirectional edges in images. An inverse Gaussian-shaped suppressing function centered at any highlighted point, which is found through the entropy filtering, is applied for restraining neighboring features in order to realize a properly distributed set of features. We can control how condense the distribution density of feature points by scaling the size of suppressing function. Finally, in sequence, many features which have rich texture cues and robustness against ill-conditions are stably extracted.

2.2 ORIENTATION CODE (F.Ullah et al., 2001, F.Ullah et al., 2002)

OC is defined as a quantized orientation of maximum intensity change in the neighbor consisting of eight pixels, of which quantization level can be arbitrarily set by providing any quantizing width. Let $\Delta I_x = \frac{\partial I}{\partial x}$ and $\Delta I_y = \frac{\partial I}{\partial y}$ represent a horizontal element and a vertical one of intensity gradient, which can be calculated by, for instance, Sobel operator (I.Sobel, 1978). By providing a quantize width Δ_θ, the code c is defined at any pixel position as

$$c = \begin{cases} \left[\frac{\tan^{-1}\left(\frac{\Delta I_y}{\Delta I_x}\right)}{\Delta_\theta}\right] & \text{if } |\Delta I_x| + |\Delta I_y| > \Gamma \\ N = \frac{2\pi}{\Delta_\theta} & \text{otherwise} \end{cases} \quad (1)$$

where N is the size of quantization, $0 \leq c \leq N-1$, and Γ is a threshold for suppressing unreliable codes generated from neighbors with low contrast, for which we prepare an exceptional value N, and we have used $\Delta_\theta = \frac{\pi}{8}$ and $N = 8$ throughout the paper.

2.3 ENTROPY FILTERING (H.Takauji et al., 2005)

In order to design stable tracking algorithms, we should consider about the quality of projected images real objects on the image plane. Since features should be prominent themselves and discriminated from other similar parts in the images, the richness in OC coded images were utilized to define these features (H.Takauji et al., 2005), which is an evaluating scheme based on a OC distribution through a modified entropy measure. Here, we evaluate the informative quality of features by the messiness of code distribution, which guarantee the inclusion of various codes in a local region. Furthermore, since the features are extracted in OC images and matched in terms of OCM, they can be robust for occlusion caused by the motion of the targets, and preferable to be insensitive to deformation due to slight change in viewing direction of the viewers or cameras (H.Takauji et al., 2005).

Let $h(i)$ $(i = 0, 1, \cdots, N-1)$ represent the number of i-th orientation codes in a M by M local region centered at any interest pixel. Then, their relative frequency is computed as

$$P(i) = \frac{h(i)}{M^2 - h(N)} \qquad (i = 0, 1, \cdots, N-1) \quad (2)$$

where $h(N)$ means the exceptional values and we exclude it from the relative frequency calculation. Next, an entropy E is defined as

$$E = \sum_{i=0}^{N-1} P(i) \log_2 P(i) \quad (3)$$

The maximum value of the entropy $E_{max} = \log_2 N$ comes of the uniform distribution, i.e., $P(i) = \frac{1}{N}$. Then, the richness or the modified entropy is defined as

$$R = \begin{cases} \frac{E - \alpha E_{max}}{E_{max} - \alpha E_{max}} & \text{if } E \geq \alpha E_{max} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where a parameter α $(0 < \alpha < 1)$ can be used to remove low entropies. Calculating the entropies and selecting prominent features is called as entropy filtering.

The entropy filtering is effective both for landmark definition and general feature definition, which are well used in many robotic systems, because we need in both cases some quick detection of positions of the candidates in observed images.

2.4 SUPRESSING FEATUERES

In order to suppress too many features which locate in crowded spaces, we introduce a suppressing function into feature extraction, which can be represented by an inverse Gaussian-like weighting function $w(x, y)$ shown in Fig 2(a).

$$w(x, y) = 1 - \frac{8}{\pi S^2} \exp\left(-\frac{(x^2 + y^2)}{S^2/8}\right) \quad (5)$$

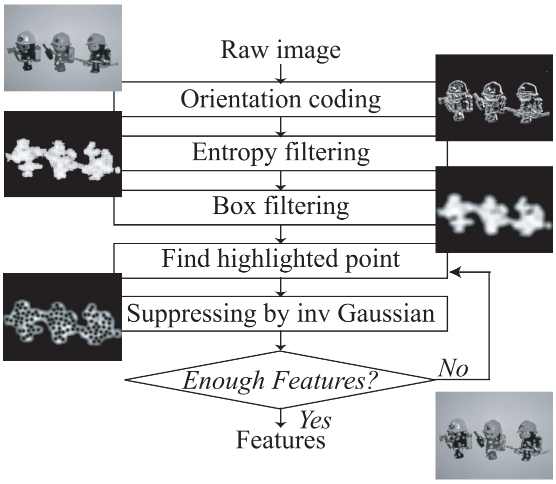

Figure 1: Feature extraction.

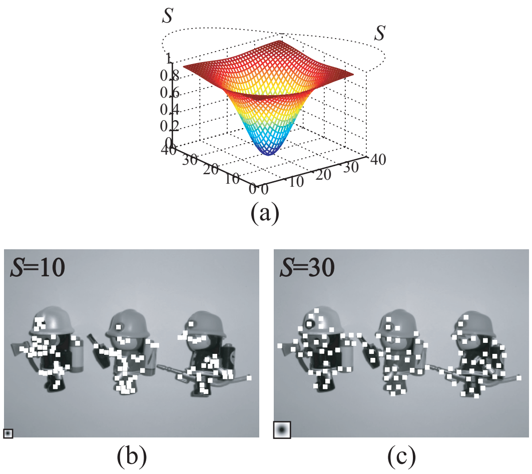

Figure 2: Suppressing function $w(x, y)$ ($S = 10$). (a) Profile of $w(x, y)$, (b) 100 features ($S = 10$), (c) 100 featues ($S = 30$).

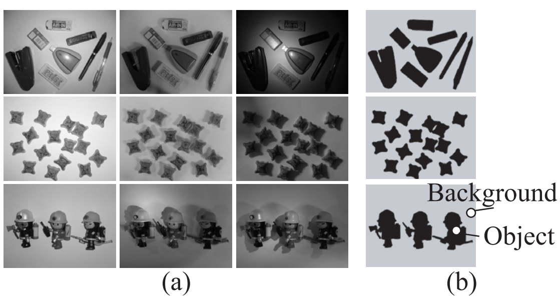

Figure 3: (a) Nine images for experiment, three scenes are taken under three different illumination conditions. (b) Pre-ordained object and background areas of the three scenes.

The iterative application of the weighting function, we can properly distribute features over the images.

We should note the effect of parameters in the procedure of the proposed feature extraction, because the performance depends on how to give the parameters, of which sensitivities are different one another. The number of OC levels N and the threshold for richness definition α affect to the richness distribution. If N becomes larger, the richness image becomes more detailed. And if α becomes smaller, the richness of all pixels in images becomes larger. But these two parameters are not changed mostly, because they do not heavily affect the result of the feature extraction. The side of the domain for entropy definition M affects the richness too, because it determines whether or not the richness image becomes blurry, and then it should be designed by the necessary resolution of the size of targets. The threshold in the definition of orientation codes Γ is thought to be insensitive to code generation, where in cases of noisy images we have to have larger Γ for restraining suspected codes due to noise effect. The size of the Gaussian-like filter S affects the density of the extracted features. If 320×240 images are used in our experiments, we often use the parameters as follows: $N = 16$, $\alpha = 0.005$, $M = 10 \sim 20$, $\Gamma = 10 \sim 150$, $S = 10 \sim 36$.

3 TRACKING

3.1 ORIENTATION CODE MATCHING (F.Ullah et al., 2001, F.Ullah et al., 2002)

In order to do tracking of the features in the real world scene, it is important to use a tough matching algorithm which should have some characteristics, easiness to use, small computation, robustness for brightness change and selectivity.We use Orientation code matching (OCM) of these characteristics, which is based on OC images. As a similarity measure between a reference image f and a target image g, the same size of $L \times L$, the mean of absolute residuals D in OC is defined as

$$D = \frac{1}{L^2} \sum_{L \times L} d(c_f, c_g) \tag{6}$$

$$d(a, b) = \begin{cases} \min\{|a - b|, N - |a - b|\} & \text{if } a, b \neq N \\ \frac{N}{4} & \text{otherwise} \end{cases} \tag{7}$$

where in the definition of piecewise absolute residual the cyclicity of orientation codes has to be taken into account. It is obvious for OCM to consist of simple procedures, resulting light computations. We know that OCM bears a narrow distribution around just neighboring region of correct matching and enough higher constancy in matching between any uncorrelated images in comparison with a famous similarity measure, for instance Normalized cross correlation (CC). Because OC has already used in the feature extraction process, the application of OCM makes it possible to design a seamless algorithm of feature extraction and then their tracking.

3.2 SPEEDING UP

In this subsection, we propose a computation cost reduction technique of affinity with the image-based tracking methods like ours. The proposed technique consists of two parts: a richness-based part and a prediction-based one. First, the richness-based part uses coherency between the reference image and target images in richness. Since the richness has rather stability against interframe motions of objects and deformations, the richness coherency between the reference in the n-th frame and any target image in the $n+1$-th frame. Define R_r^n as the richness of the reference in the n-th frame, while R_t^{n+1} as a target image in the $n+1$-th frame, and then, a set or regions of candidate positions for precise matching in the $n+1$-th frame, Ω_R^{n+1} can be defined as follows:

$$\Omega_R^{n+1} = \{Vp = (x, y) | \, |R_r^{n+1}(Vp) - R_t^n| \leq \epsilon\} \tag{8}$$

where ϵ is an arbitrary constant and determines the allowable range of variation in the richness. We can limit search in the $n+1$-th frame to the positions in Ω^{n+1} for speeding up. Precision and cost reduction in the tracking depend on ϵ, so we have to make them balanced in some reasonable situation, because ϵ should be small to put the cost reduction ahead of the precision and vice versa.

Second, the information of feature location is used to decimate the other areas. We can predict locations of target images in the $n+1$-th frame by use of a Kalman filter and the previous positions of them in the n-th frame. The candidate positions p^{n+1} can be included in a set Ω_P^{n+1} by letting $\widehat{p}^{n+1/n}$ the predicted location of any target image in the $n+1$-th frame and providing ω as a radius of prediction as follows:

$$\Omega_P^{n+1} = \{p_t^{n+1} | \, |p_t^{n+1} - \widehat{p}^{n+1/n}| \leq \omega\} \tag{9}$$

But we sometimes not so small errors in the prediction provided by a Kalman filter, which needs many kinds of preparation to realize for practical use.

Therefore, we alternatively can utilize an assumption of stationarity in which any sudden change of the position of targets never happens, resulting that predicted locations of target images are very near to the location of the previous position. Let p_r^n the location of the reference image in the n-th frame. Then, a set Ω_S^{n+1} of candidate positions p_t^{n+1} is defined as

$$\Omega_S^{n+1} = \{p_t^{n+1} | \, |p_t^{n+1} - p_r^n| \leq \omega\} \tag{10}$$

By using both the richness-based part and the prediction-based part, the conclusive candidate region Ω^{n+1} is defined as

$$\Omega^{n+1} = \Omega_R^{n+1} \cap (\Omega_P^{n+1} \cup \Omega_S^{n+1}) \tag{11}$$

A simpler version of it can be defined as

$$\Omega^{n+1} = \Omega_R^{n+1} \cap \Omega_S^{n+1} \tag{12}$$

In the experiments mentioned in the following sections, the results of (12) have been explained.

4 FUNDAMENTAL EXPERIMENTS

4.1 FEATURE EXTRACTION IN COMPARISON WITH KLT

In this subsection, the performance of the feature extraction algorithm is evaluated in comparison with a well-known extraction algorithm. To evaluate performance of the feature extraction method, the comparison between our method and KLT (Kanede-Lucas-Tomashi) (J.Shi and C.Tomasi, 1994) method have been tried. KLT is a well-known feature extraction and tracking method and is based on a model of affine image change. It is optimized for feature extraction from video-sequences. The comparison of feature extraction methods for their performance evaluation have been tried by many researchers. Tankus (A.Tankus and Y.Yeshurun, 2005) evaluates feature extraction methods by computing the actual correct track time of a track. Schmid (C.Shimid et al., 2000) uses Repeatability of feature extraction positions between two different scenes and information Content. KLT shows high performance in both experiments. The implementation details of KLT based on a free program in C++ code being made public on the Internet [1]. Parameters of our method are set as $N = 17$, $\alpha_e = 0.995$, $M = 10$, $\Gamma = 17$, $S = 23$.

Fig. 3 shows the scenes for the experiments, which include different kinds of objects, stationeries, cookies and toys. They are taken under three different illumination conditions. The object area (black) and the background area, a white flat plane (gray), are defined by Fig. 3 (b). One hundred features have been extracted from each image by our method and KLT. Examples of extraction results are shown in Fig. 4. For example, KLT extracts many features from the shadow as illustrated in Fig. 4, where false features are mainly from edges and corners. We have checked a feature detection rate from the object areas. In these experiments, we could obtain by our method the detection rate 95.8%, while by KLT the ratio could be 86.5%. This result can be explained by the fact that RC is robust for brightness change caused by illumination fluctuation but KLT is not so.

4.2 APPLICATION TO PLANER LANDMARKS TRACKING

In order to do performance survey and evaluation of the OCM tracking algorithm, we have tried tracking planer landmarks. Five pieces of landmark of star-and-circle shape are depicted on a plate as shown in Fig. 5 (a), and they are subject to be tracked in the scene with change in posture, scaling and illumination. The landmarks have high richness, therefore they are extracted easily by using texture richness analysis. The plate is moved manually by an operators hand. Fig. 5 (b) shows the result of the plain landmark tracking. The areas which are enclosed by squares are tracked target areas. Instead of the deformation of the target areas, rotation, scaling and illumination change in the motion of the plain object, the method can keep tracking well. The parameters are set as below. $N = 16$, $M = 20$, $\Gamma = 23$, $\alpha_e = 0.88$. The image size is 320×240. The frame rate of the algorithms is about 7.8fps (Pentium4 1.8GHz, 512MB RAM).

5 DISCUSSION

5.1 AUTOMATIC TEMPLATE DEFINITION

If objects which should be tracked have no landmarks, we have to define templates (or RoI) for tracking automatically. The features being extracted by our method are well-suited to tracking, but too much to track. Therefore, we need to define few templates from this feature distribution. In the previous chapter, we define templates on man's face by detecting areas which have highest Richness. But it does'nt seem to be with wide application. In this chapter, we propose an automatic template definition method and show results of tracking experiments using the method. K-means (J.B.MacQueen, 1967), a well known and simple clustering algorithm to group features and calculate their centroids, is used for this purpose. The features have been classified into a specified number of clusters, templates can be adopt to the cluster centroids. Grouping extracted features, their position information is used. Another adequate data therefore could be the richness values. The first step of the k-means algorithm is to define k initial centroids, this number is fixed and depends on the quantity of

[1] http://robotics.stanford.edu/˜birch/klt/

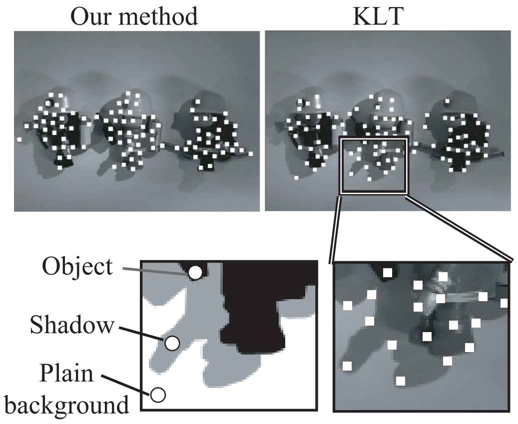

Figure 4: Examples of extraction results, KLT extracts many features from shadows of objects.

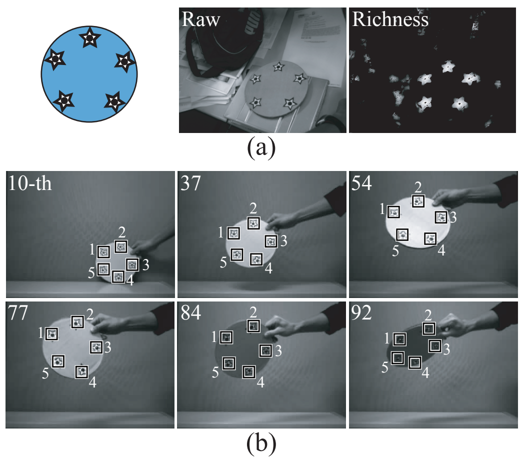

Figure 5: Planar landmark tracking using OCM, (a) An object having six high richness star-shape landmarks, (b) a tracking result.

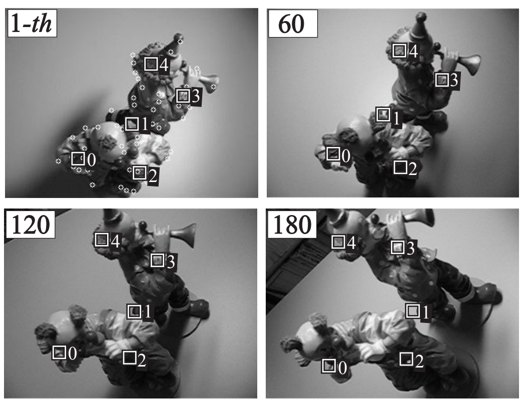

Figure 6: The result of dolls tracking using automatic template generation and update.

features. We set these uniformly distributed near the mid of the raw video data in the first frame, so that we can provide, that the initial templates will fit with tracked objects. Templates and centroids respectively outside the center often doesn't conform with features, and become useless. But to prevent this, centroids could be specified also by coincidence within a segmented area. Then, for each feature, the distance to these (initial) centroids is computed and the feature is assigned to the closest one. When all features have been assigned, the centers of the k clusters have to again be determined. The last two steps must be repeated until the centers do not move any longer. To measure the distance between a feature $f_i^{(j)}$ and a centroid c_j, we uses the Pythagorean theorem. K-means algorithm aims at minimizing a function, that is defined as

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \sqrt{\left(f_{i,x}^{(j)} - c_{j,x}\right)^2 + \left(f_{i,y}^{(j)} - c_{j,y}\right)^2} \qquad (13)$$

This function is an indicator of the distance of the n features to their associated cluster centers. It is proved (T.Kanungo et al., 2000) that k-means algorithm will always terminate, but it does not necessarily find the most optimal result, analogous to the global function minimum. But we obtained thereby sufficiently good results and it is not necessary to run the clustering procedure several times. But in uncommon cases it is possible, that we create cluster without or only one feature, then we do not use them for further activities. The complicacy is how to choose k. In our experiments, this parameter is set manually, for an automatic localization background knowledge is essential. Additive information like the number and size of persons (in relation to the template size) in a scene or even the ken of segmented areas could be very helpful. An advancement of performance with k-means using background knowledge is described in (K.Wagstaff et al., 2001). An optimal value of k is also dependent on n. A disadvantage of the presented method is the adverse effect by features they are not related to the interesting object. Only a few features, detected in the background, could cause movement of cluster centroids to an area in the video sequence that not contains interesting objects to track. To prevent this, a real-time background subtraction is needed. But the way to distinguish between objects and background depends on applications. Therefore, we don't make further reference. To give an example, moving man's face tracking under complicated background have been tried in (Y.Domae et al., 2005) and forward subtraction is used in that algorithm to select features on the moving face. Above-mentioned we described a method to initialize templates by finding well suited positions with clustering by k-means. The templates are captured at the cluster centroids in the initial frame and tracked in the following by OCM.

5.2 AUTOMATIC TEMPLATE UPDATE

Feature extraction and clustering is not necessary for the tracking run, except some special cases. Such cases can be movements of objects (e.g. persons) out of the regarded scene, or head turn of a human that occurs occlusions, or even an intense illumination change like switching the light off (contrast becomes too low). Under such circumstances occurs an error because the template can not be tracked any longer. But it is not possible to identify such failures by a threshold or something else relative to the OCM values, because these values are always unsteady and nearby. So we decided to use the displacement of the central point for each template. If the distance of centers in consecutive frames exceed a defined threshold Γ we realize a misplaced template and know that we have to update this one. A new feature extraction and cluster recalculation is necessary. It is conceivably to use the last known template positions for cluster initialization, but there is no strong discrepancy ascertainable. The updated template is extracted at the appropriate cluster centroid, for remaining templates nothing changes. We tried a lot of experiments under various conditions, with object and camera movings, zoom in and out, occlusions of tracked sections, changes in illumination, a different number and size of templates, and so on.

5.3 APPLICATION TO SCENES WITH DOLLS

We made a lot of experiments using dolls as tracked objects and tried camera moving and examine different parameter settings like above-mentioned. An example is shown in Fig. 6.

By researching scenes with camera movings and shadowing we didn't assert updates evoked by this proceeding. Only if a tracked part of a doll moved out the regarded scene, an updating process is executed. The same effect is visible, if the light is switched off and the contrast become too low in two consecutive frames. In all probability an update is necessary. If there is an update in progress and the scene changes two much since the initialization or the last redefinition frame, especially in the background, it is possible that the new template is set to a undesirable position that is not related to the previous tracked section. In these scenes we obtained best results by using 25 up to 100 features, 3-5 templates of size $15 \times 15px$. There is no advantage by using more then 100 features.

5.4 COMPARISON WITH CC

For evaluating the template updating process, the matching error of template matching methods can be used. If we compare the number of updated templates between OCM and CC in the same video sequences, we can understand which method is superior in robustness against illumination change or deformation of objects. While doing the experiments, we have counted the number of updated templates to become an idea of quality of the tracking algorithm. The average of the update rate by using OCM for tracking is lower than 2%, which is a very good result. Nearly all redefinitions are caused by occlusions, discontinued tracking by object moving out the regarded scene or the total losing of contrast between two consecutive frames by switching light off. Slower brightness changes can cope with OCM tracking. Unlike the Correlation Coefficient (CC) method: the update rate becomes about 5%, there are redefinitions without distinguishable reasons. In both methods, the quality of the raw material influences the results. So we obtained best update rates by using seperated bitmap-files of each frame and not that good results by capturing images from a webcam input.

6 CONCLUSION AND FUTURE WORKS

We have described a novel algorithm for feature extraction and tracking, which consist of OC, Richness evaluation, suppression by using inverse Gaussian, OCM and automatic template definition using k-means. And then, the robustness of the algorithm against change in illumination was shown formally and experimentally. The effectiveness of the algorithm is tested and recognized through comparison the algorithm to KLT. The landmark and dolls tracking have been performed for evaluation of the proposed algorithm.

As future works, we are going to improve the algorithm in computation cost and robustness, and as an important research topic some formalization of tracking and template modification will have to be considered with experiments using real world data.

REFERENCES

A.Tankus and Y.Yeshurun, 2005. Scene-consistent detection of feature points in video sequences. Computer Vision and Image Understanding 97(.), pp. 1–29.

C.Harris and M.Stephens, 1988. A combined corner and edge detector. In: Proc. 4th Alvey Vision Conf.., pp. 147–151.

C.Shimid, R.Mohr and C.Bauckhage, 2000. Evaluation of interest point detectors. International Journal of Computer Vision 37(2), pp. 151–172.

D.Tweed and A.Calway, 2002. Tracking many objects using subordinated condensation. In: Proc. of the British Machine Vision Conf., pp. 283–292.

F.Ullah, S.Kaneko and S.Igarashi, 2001. Orientation code matching for robust object search. IEICE Trans. of Inf. & Sys E84(D(8)), pp. 999–1006.

F.Ullah, S.Kaneko and S.Igarashi, 2002. Tracking looming and receding objects by orientation code matching. The Journal of the IIEEJ 31(1), pp. 94–102.

H.Takauji, S.Kaneko and T.Tanaka, 2005. Robust tagging in strange circumstance. The Transactions of the Institute of Electrical Engineers of Japan (C) 125(6), pp. 926–934. (in Japanese).

I.Sobel, 1978. Neighbourhood coding of binary images for fast contour following and general array binary processing. Computer Graphics and Image Processing 8(.), pp. 127–135.

J.B.MacQueen, 1967. Some methods for classification and analysis of multivariate observations. In: Proc. Fifth Berkeley Sympos. Math. Statist. and Probability, pp. 281–297.

J.Canny, 1986. A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 8(6), pp. .

J.Shi and C.Tomasi, 1994. Good features to track. In: Proc. IEEE Conf. Comput. Vision Patt. Recogn, pp. 593–600.

K.Wagstaff, C.Cardy, S.Rogers and S.Schroedl, 2001. Constrained k-means clustering with background knowledge. In: Proc. 18th International Conference on Machine Learning, pp. 577–584.

M.Isard and A.Blake, 1996. Contour tracking by stochastic propagation of conditional density. In: Proc. of European Conf. Computer Vision, pp. 343–356.

R.E.Kalman, 1960. A new approach to linear filtering and prediction problems. In: Trans. of the ASME-Journal of Basic Engineering, pp. 35–45.

T.Kanungo, D.Mount, N.S.Netanyahu, C.Piatko, R.Silverman and A.Y.Wu, 2000. Analysis of a simple k-means clustering algorithm. In: Proceedings of Symposium on Computational Geometry 2000, pp. 100–109.

Y.Domae, S.Kaneko and T.Tanaka, 2005. Robust tracking based on orientation code matching under irregular conditions. In: Proc. of SPIE, pp. 60510S–1–60510S–9.

Y.Lamdan and H.J.Wolfson, 1977. Towards automatic visual obstacle avoidance. In: Proc. 5th Int. Joint Conf. Art. Intell, p. 584.